LLNL-TR-410855

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

GT Merge Process: Version 1.0

M. P. Flanagan, D. Dodge, S. C. Myers

February 26, 2009

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

GT Merge Process: Version 1.0

Megan P. Flanagan, Doug Dodge, and Stephen C. Myers

Table of Contents

Introduction

This document summarizes the process used to merge GT25 and better data between LANL and LLNL for use in a tomographic inversion for *Pn* velocity of Eurasia. The merge process is automated and includes extensive quality control operations at each step. Events in common between the labs are identified and resolved using GT level criteria. Arrivals in common between the labs are also resolved through the use of agreed upon arrival author rankings. Finally, baselined origin times are computed for all crustal events using either teleseismic *P*-arrivals and the *iasp91* model or, in certain regions, regional *P*-arrivals and regional velocity models that are known to be consistent with teleseismic *iasp91 P*-wave predictions.

Installation Steps

The results of the GT merge are contained in an Oracle 9 dump file named *gtmerge.dmp*. This file should be imported into an empty schema. Failure to do so will likely cause data corruption and make unusable both the merge data and any data that existed in the schema prior to the merge. The objects in the dump file were created in the table space GTMERGE_DATA and indexes were created in the table space GTMERGE_INDEX. Therefore, these table spaces must exist and have sufficient space available for the import to succeed. The GTMERGE_DATA table space must be 6.0 Gbytes and the GTMERGE_INDEX must be 7.5 Gbytes.

Executive Summary

We combine the core tables from each lab and first resolve unique and common GT events between LANL and LLNL. Phase names are then checked and possibly adjusted for consistency. Next, we merge at the pick level so that each distinct *EVENT-STATION-PHASE* tuple has a unique arrival. All BMEB (Bondar-Myers-Engdahl-Bergman) GT are evaluated for adherence to their criteria, and possibly re-calculated. Finally, new origin times are computed (baselining) for the merged GT events.

In addition to the reconciliation of events and picks between LANL and LLNL, the merge process involves several quality control steps that are intended to remove outlier and irrelevant data from the final results.

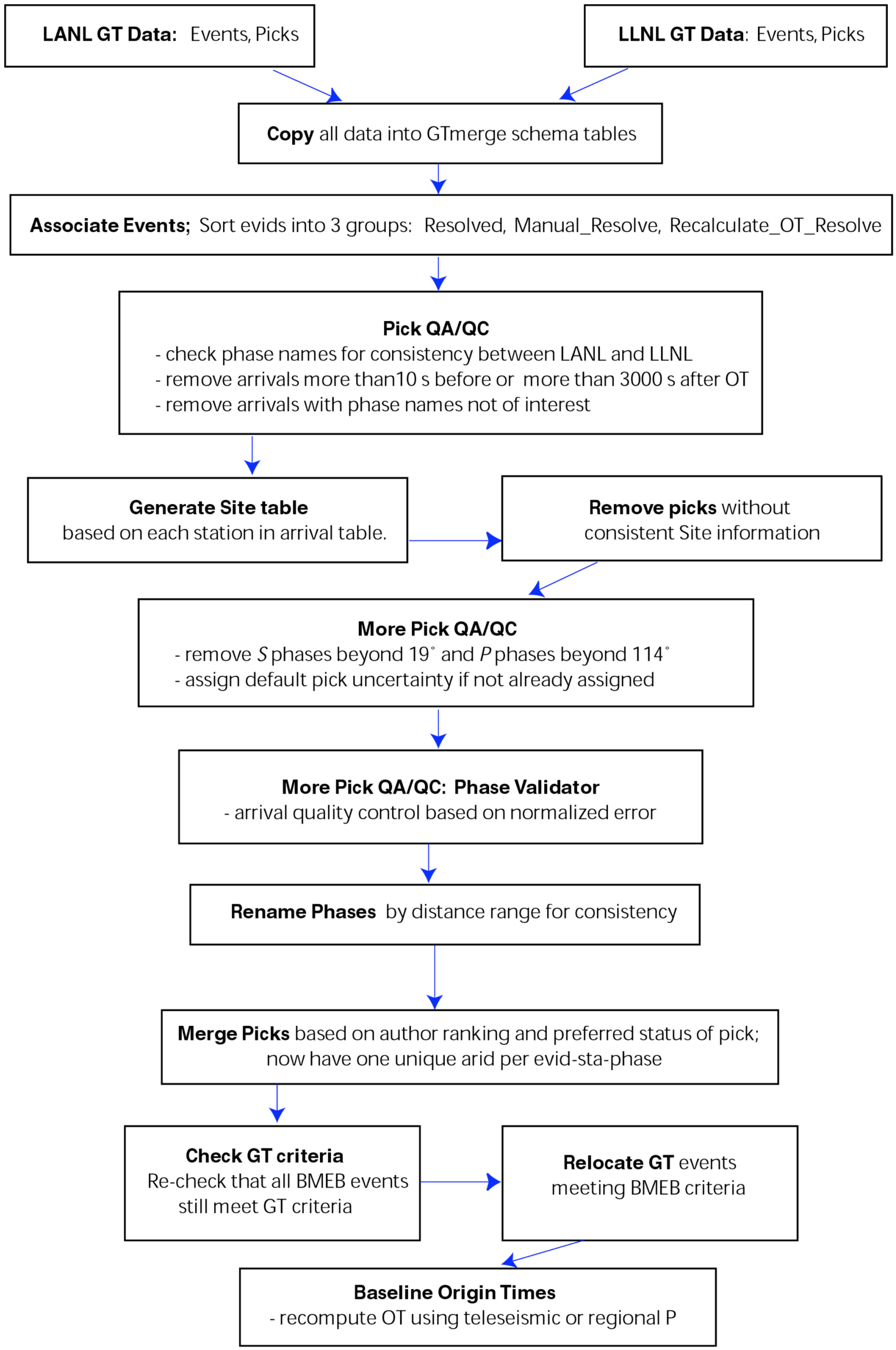
The merge steps are described broadly in this summary and in more detail in Appendix 1. The major steps are:

- Gather GT data from all relevant LANL and LLNL tables and copy into the GTMERGE schema. Note that merging of events only takes place in a merge region defined for the *Pn* tomography effort as the geographic range of latitude 0° to 90° and longitude -20° to 150°. However, for simplicity and data space considerations, LLNL provided GT data for the entire Earth as the merge codes drive off the full LLNL schema and table contents. Therefore, results are presented both for the merge region and for the entire Earth.
- Identify events in common between the labs, and for such events choose one of the two competing GT solutions based on the GT level and criterion type. At this stage GT of type BMEB (Bondar-Myers-Engdahl-Bergman) are not resolved. Instead, their *evids* are moved to the RECALC_ORIGIN_RESOLVE table because all such GT are subject to re-

calculation after QC of arrivals. All associated arrivals are also given entries in the EVENT_ARRIVAL_ASSOC table. This grouping of arrivals by *evid* rather than by one origin solution (*orid*) facilitates recalculation of BMEB GT.

- For LLNL events, retrieve additional arrivals that are associated with the events in the GTMERGE schema but which are not associated with a GT origin. These are loaded into ARRIVAL_EX and associated with the appropriate event through the EVENT_ARRIVAL_ASSOC table.
- Phase names are checked for consistency between labs (*e.g.*, *P** is renamed to *Pb*), phases with arrival times more than 10 s before or 3000 s after origin time are removed as a first cut, and phases not found in the agreed upon ALLOWABLE_PHASE table are removed.
- The SITE table is built based on all stations listed in ARRIVAL_EX table. For arrivals with no site data, the station code (along with an arrival count) is written to the MISSING_SITE table.
- Phases outside usable distance ranges are removed and pick uncertainty (*deltim*) is assigned if not already set.
- Arrivals for stations in the MISSING_SITE table are removed.
- The quality of each arrival is assessed by evaluating the travel-time residual normalized to expected fluctuations in travel-time prediction and measurement error. Low-quality arrivals are removed.
- All phases (*P, Pg, Pn, S, Sg, Sn, Lg*) in the ARRIVAL_EX and EVENT_ARRIVAL_ASSOC tables are checked and possibly renamed by distance range.
- Merge arrivals for each *EVENT-STATION-PHASE* based on author ranking and preferred status of the pick; this yields a unique *arid* for every *evid-sta-phase* tuple.
- Check that BMEB GT *evids* meet appropriate GT criteria; for every BMEB-GT event, check that the associated arrivals all meet the GT distance and azimuth criteria. If not, then move the *evid* to the RECALC_ORIGIN_RESOLVE table.
- Relocate all BMEB events in RECALC_ORIGIN_RESOLVE. If the new location meets BMEB criteria, then write new ORIGIN_EX, ASSOC, PREDICT_TT, and GT_EPI_EX rows and move *evid* to RESOLVED, if not, write summary to FAILED_BMEB_RECALC and FAILED_BMEB_RECALC_DETAIL tables.
- Baseline origin times. For all *orids* in the RESOLVED table, re-compute the origin time (constrained origin solution) using only teleseismic *P* phases (must have minimum of 10 *P* phases) and write new ORIGIN_EX, ASSOC, PREDICT_TT, and CAL_ORIGIN rows.
- For events in the European Arctic, western China, and YSKP regions, not baselined in the previous step, we use appropriate regional models to perform baselining and write new ORIGIN_EX, ASSOC, PREDICT_TT, and CAL_ORIGIN rows.

A Flowchart for Merge Processing Steps is illustrated on the next page**:**


LANL GT Data: Events, Picks
LLNL GT Data: Events, Picks
Copy all data into GTmerge schema tables
Associate Events; Sort evids into 3 groups: Resolved, Manual_Resolve, Recalculate_OT_Resolve
Pick QA/QC
- check phase names for consistency between LANL and LLNL
- remove arrivals more than10 s before or more than 3000 s after OT
- remove arrivals with phase names not of interest
Generate Site table
based on each station in arrival table.
Remove picks without
consistent Site information
More Pick QA/QC
- remove S phases beyond 19° and P phases beyond 114°
- assign default pick uncertainty if not already assigned
More Pick QA/QC: Phase Validator
- arrival quality control based on normalized error
Rename Phases by distance range for consistency
Merge Picks based on author ranking and preferred status of pick;
now have one unique arid per evid-sta-phase
Check GT criteria
Re-check that all BMEB events
still meet GT criteria
Relocate GT events
meeting BMEB criteria
Baseline Origin Times
- recompute OT using teleseismic or regional P

Merge Results

At the completion of the merge there were a total of 86504 GT events. Of these, 53438 were within the merge region and 19719 were within the merge region and also had baselined origin solutions. A break-down of these numbers by GT level is shown in the tables below.

Table 1. Final GT epicenter data for Entire Earth.

GT Level	Number
25	35354
5	30384
15	8824
20	8582
2	2061
0	996
1	275
3	26
0.001	1
0.1	1

Table 2. Final GT epicenter data for merge region.

GT Level	Number
5	21673
25	15998
15	7148
20	6198
2	2061
1	275
0	57
3	26
0.001	1
0.1	1

Table 3. Final GT epicenter data in merge region with baselined origins.

GT Level	Number
25	10657
15	5126
20	1718
2	949
5	939
1	262
0	51
3	16
0.1	1

Fate of Contributed GT

The two labs contributed a total of 118302 GT_EPI rows. Of these, 21954 were contributed by LANL and were inside the merge region: latitude 0° to 90° and longitude -20° to 150°. The LLNL contribution consisted of 92991 rows for the entire Earth. After merging the data, 97400 unique events were identified. Not all of these events survived to the end of the merge because in many cases the BMEB GT were found not to meet their criteria.

Before removal of bad GT there were:

- 20816 events in common
- 72089 events by LLNL only
- 4495 events by LANL only

The raw total of 97400 GT was later reduced to 86504 because 10896 BMEB GT solutions did not meet their criteria after arrival cleanup and phase validation.

Relevant SQL:

Get events in Common:

```
(select evid from gt_epi_ex where contrib_org = 'LLNL'
union
select evid from removed_gt_epi where contrib_org = 'LLNL')
intersect
(select evid from gt_epi_ex where contrib_org = 'LANL'
union
select evid from removed_gt_epi where contrib_org = 'LANL');
```

Get LLNL-only Events:

```
(select evid from gt_epi_ex where contrib_org = 'LLNL'
union
select evid from removed_gt_epi where contrib_org = 'LLNL')
minus
(select evid from gt_epi_ex where contrib_org = 'LANL'
union
select evid from removed_gt_epi where contrib_org = 'LANL')
```

Get LANL-only Events:

```
(select evid from gt_epi_ex where contrib_org = 'LANL'
union
select evid from removed_gt_epi where contrib_org = 'LANL')
minus
(select evid from gt_epi_ex where contrib_org = 'LLNL'
union
select evid from removed_gt_epi where contrib_org = 'LLNL')
```

Get removed GT Events:

```
select distinct evid
  from removed_gt_epi
 where reason = 'BMEB solution removed for (failed) re-calculation.'
```

At the end of the merge there were 86504 unique events, spanning the whole globe, with acceptable GT remaining. Of these,

- 66086 had only a LLNL contribution.
- 1747 had only a LANL contribution.
- 18671 had a contribution from each lab.

Relevant SQL:

```
select contrib_org, count(*) from gt_epi_ex group by contrib_org;
```

Within the merge region there were 53438 unique events. Of these,

- 33420 had only a LLNL contribution,
- 1747 had only a LANL contribution, and
- 18271 had a contribution from each lab.

Note that there are 400 fewer events in common between the labs within the merge region than in total. The missing GT are BMEB that fall outside the merge after re-calculation.

Relevant SQL:

Get contributor totals inside merge region:

```
select contrib_org, count(*) from gt_epi_ex where lat between 0 and 90 and lon between -
20 and 150 group by contrib_org;
```

Get GT type for merge events not inside merge region:

```
select method, count(*)
  from gt_epi_ex
 where evid in (select evid
                  from gt_epi_ex
                 where contrib_org = 'merge'
                minus
                select evid
                  from gt_epi_ex
                 where contrib_org = 'merge'
                   and lat between 0 and 90
                   and lon between - 20 and 150)
 group by method
```

Out of the 86504 GT events, there were a total of 33965 baselined origin solutions computed. There were 24046 events that were too deep to be used and 28493 events did not have enough observations for teleseismic baselining and which could not be baselined using regional phases. Within the merge region there were 19719 baselined events. Of these,

- 11621 had only a LLNL contribution,
- 1328 had only a LANL contribution, and
- 6770 had a contribution from each lab.

Relevant SQL:

Get total baselined GT:

```
select count(*) from cal_origin
```

Get counts on failed baseline GT:

```
select reason, count(*) from unbaselined_gt group by reason
```

Get contributor totals for baselined GT inside merge region:

```
select a.contrib_org, count(*)
  from gt_epi_ex a, cal_origin b
 where a.evid = b.evid
   and lat between 0 and 90
   and lon between - 20 and 150
 group by a.contrib_org
```

LLNL Contributions

- 92991 rows at start
- 189 removed because the LANL contribution had a better GT level
- 86 removed and replaced by a row with CONTRIB_ORG called '*merge*' because the two lab's contributions were essentially identical.
- 2 removed because the data were inconsistent between labs and the region owner's data was retained.
- 26628 removed for re-calculation of the BMEB solution. Of these:
    - 18296 were successfully re-calculated
    - 8332 failed to meet criteria after re-calculation.

LANL Contributions

- 21954 rows at start
- 3271 removed because the LLNL contribution had a better GT level
- 86 removed and replaced by a row with CONTRIB_ORG called '*merge*' because the two lab's contributions were essentially identical.
- 2 removed because the data were inconsistent between labs and the region owner's data was retained.
- 20205 removed for re-calculation of the BMEB solution. Of these:
    - 17215 were successfully re-calculated
    - 2990 failed to meet criteria after re-calculation.

The relevant SQL is:

```
select contrib_org, reason, count(*)
  from removed_gt_epi
 group by contrib_org, reason;
```

Similarly, a breakdown of the 3271 LANL removed events by *gtlevel_km* is given below, showing that the most were GT20 and GT25 compared to the LLNL GT15 level.

Table 4. 3271 LANL *evids* removed due to better LLNL GT.

LLNL		LANL		
gtlevel_km	**method**	**gtlevel_km**	**method**	**count**
15	EHB50/90	25	Bondar et al. (2004) criteria	2144
15	EHB50/90	20	Bondar et al. (2004) criteria	632
5	Engdahl_Bergman	15	cluster analysis	156
3	skorve_pub	5	general explosion	13
1	sultanov_pub	5	general explosion	2
0	refraction shot	5	general explosion	1
3	skorve_pub	25	Bondar et al. (2004) criteria	1
5	InSAR	25	Bondar et al. (2004) criteria	1
1	satellite image	5	satellite shaft locations, JHD locations	1
2	pidcgt0	5	general explosion	1

A breakdown of the 189 LLNL removed events by *gtlevel_km* is given below.

Table 5. 189 LLNL *evids* removed due to better LANL GT.

LANL		LLNL		
gtlevel_km	**method**	**gtlevel_km**	**method**	**count**
1	SPOT satellite image, teleseismic epicenter est.	5	Engdahl_Bergman	43
5	general explosion	15	50/90	30
1	known locations, map est. Degelen adit complex	5	Engdahl_Bergman	26
0	known locations, map est. Degelen adit complex	5	Engdahl_Bergman	14
5	general explosion	25	sultanov_pub	14
15	cluster analysis	25	BMEB-GT25	12
1	IKONOS satellite imagery, seismogram alignment	5	Engdahl_Bergman	8
1	known locations, map est. Degelen adit complex	5	BMEB-GT5	5
0	SPOT satellite images, teleseismic epicenter est.	5	Engdahl_Bergman	4
1	known nuclear explosion	3	skorve_pub	4
0	known locations, map est. Degelen adit complex	5	BMEB-GT5	4
1	SPOT satellite images, teleseismic epicenter est.	5	BMEB-GT5	3
1	IKONOS satellite imagery, seismogram alignment	5	BMEB-GT5	2
2	double-difference analysis of regional/tele data	5	Engdahl_Bergman	2
5	teleseismic relocation, depth determination	15	50/90	2
5	InSAR and seismic methods	25	BMEB-GT25	2
5	InSAR and seismic methods	5	BMEB-GT5	2
15	cluster analysis	20	BMEB-GT20	2
0	SPOT satellite images, teleseismic epicenter est.	1	sultanov_pub	1
5	Bondar et al. (2004) criteria	15	50/90	1
3	mine collapse	5	Kremenet_pub_2001	1
1	double-difference analysis of regional/tele data	5	BMEB-GT5	1
2	double-difference analysis of regional/tele data	5	BMEB-GT5	1

Arrivals

A total of 20,259,463 arrivals were contributed by the two labs. Of these, 2,932,567 arrivals were removed for QC reasons. Many *evid-sta-phase* tuples had multiple arrivals (primarily bulletin picks contributed by both labs), so after selecting preferred arrivals for each *evid-sta-phase* there were 10,604,896 remaining at the completion of the merge. Table 4 shows the break-down by phase.

Table 6. Unique arrivals for all GT events, global.

Phase	Count
P	6,456,395
Pn	1,884,000
Sn	619,159
Pg	564,327
Sg	440,567
pP	268,531
PcP	167,242
Lg	105,519
sP	96,399
sS	1,428
PmP	546
S	289
Pb	279
Sb	194
Pdif	19
SmS	1
pS	1

For events within the merge region, there were 6,379,926 unique arrivals. Table 5 shows the break-down for this group by phase.

Table 7. Unique arrivals for all GT events inside merge region.

Phase	Count
P	3,628,690
Pn	1,263,166
Sn	472,484
Pg	329,959
Sg	295,162
pP	152,029
Lg	88,393
PcP	87,166
sP	60,401
sS	1,303
PmP	540
Pb	232
S	223
Sb	171
Pdif	7

For events within the merge region for which a baselined origin time was calculated there were 3,552,832 unique arrivals. The break-down by phase is shown in Table 6.

Table 8. Unique arrivals for all GT events inside merge region with baselined origin times.

Phase	Count
P	2,544,438
Pn	563,394
Sn	202,411
pP	79,554
PcP	56,704
sP	39,479
Pg	25,916
Sg	19,781
Lg	19,547
sS	1,110
S	176
Pb	174
Sb	134
PmP	9
Pdif	5

Manual Resolution Events

Four events had to be manually resolved because they had identical gt levels but differed in position by more than the limit of the gt level. These were resolved by choosing the solution from the lab responsible for the region in which the event occurred. Specifically:

EVID 16662 used LANL solution
EVID 53958 used LANL solution
EVID 6755 used LLNL solution
EVID 54320 used LLNL solution

Viewing Merge Results

As an aid in visualizing the merge results, we have produced two .KML files that can be viewed using the Google Earth client. The file gt_epi.kml contains all 86500 GT events with symbol style corresponding to the different gt levels. The file cal_origin.kml contains just the subset of gt for which baseline origin solutions were computed. If you have the *Google Earth Client* installed on your system you can double click the .kml files to view them.

SQL Exploration

Appendix 2 contains a description of the schema containing the merge data. For information required to construct detailed queries, please see the appendix. However, the following query is an example of extracting travel-time information for all baselined origins and associated arrivals (*e.g.*, to make an input file for *Pn* Tomography):

```
select  a.evid,
      a.orid,
      d.arid,
      d.sta,
      d.iphase phase,
      d.time-b.time obs_traveltime,
      d.deltim,
      b.lat evla,
      b.lon evlo,
      b.depth,
      e.lat stla, e.lon stlo, e.elev
from cal_origin a, origin_ex b, event_arrival_assoc c,
arrival_ex d, site e
where a.orid = b.orid
   and b.evid = c.evid
   and c.arid = d.arid
   and d.is_preferred = 'y'
   and d.sta = e.sta and d.jdate between e.ondate and e.offdate
```

This is created as the GTMERGE.TRAVELTIME_VIEW in the GTMERGE schema.

Caveats, Known Issues, and Proposed Solutions

1. Deep events were not baselined. There were about 24046 events in the RESOLVED table that had depths greater than 35 km, so these were excluded from baselining. If all or some fraction of these events are to be used then a strategy must be developed for handling deep events.

Merge Process Details

First, data from the LANL and LLNL core tables (*e.g.*, ARRIVAL, ASSOC, GT_EPICENTER, GT_DEPTH, GT_TIME, ORIGIN, ORIGERR, SITE) are copied into the GTMERGE schema target tables and remap columns are created as well (the ending _EX indicates an extended table with remap columns at the end). This is followed by the merge step in which *evids* are assigned to the data and each *evid* is placed in one of three tables: RESOLVED, RECALC_ORIGIN_RESOLVE, or MANUAL_RESOLVE.

For events in common, the better GT solution is selected by comparing first GT-level and then GT-criterion type. Currently, the only rule for criterion types is that non-BMEB always trumps BMEB GT. So, for example, in the case where two events have the same *gtlevel_km* but one is of type BMEB and the other is 'InSAR', the InSAR GT will be retained. When both lab's GT are of type BMEB and the *gtlevel_km* values are the same, then the *evid* is written to the RECALC_ORIGIN_RESOLVE table. At a later step in processing, combined arrival data will be used to estimate new GT solutions for these events and a new *auth* called '*merge*' is given to them. Any remaining common events for which the *gtlevel_km* is the same yet neither *method* is BMEB are written to MANUAL_RESOLVE and must be reviewed by hand. For those events in common that have the same *gtlevel_km* and for which neither method is BMEB, the positions are checked to see if they are within the *gtlevel_km* of one-another. If so, the first is retained and a new *auth* called '*merge*' is assigned. Otherwise, the *evids* are written to the MANUAL_RESOLVE table.

Next, we load additional arrivals from the LLNL.ARRIVAL table and into ARRIVAL_EX. These are picks that have been made for merge events but that are not yet linked to any specific *orid* in the GTMERGE schema. They do not have an entry in the ASSOC table (they are linked to the appropriate *evid* through EVENT_ARRIVAL_ASSOC). We now have a complete compilation of arrivals, and we apply some preliminary QC procedures to these phase picks. Phase names are checked for consistency between labs (*e.g.*, *P** is renamed to *Pb*), phases with arrival times more than 10 s before or 3000 s after origin time are assumed to be erroneous or extraneous and are removed. Also, phases not found in the agreed upon ALLOWABLE_PHASE table are removed at this time; any removed phases are written to the REMOVED_ARRIVAL table.

The SITE table is then built from the *sta* for every row in ARRIVAL_EX. Our SITE information is taken from an experimental SITE merge that brings together data from NEIC, ISC, LANL, IRIS and LLNL SITE tables. All but three stations involving 24 arrivals are accounted for using this SITE table. These missing stations are listed in the MISSING_SITE table. Construction of the merged SITE table is discussed in greater detail in the Appendix 3.

More arrival QC is now performed on the ARRIVAL_EX table to remove unwanted phase types not necessary for the merge. Specifically, each allowable phase arrival with *deltim* ≤ 0.0 or which is a LANL arrival with a LANL default value for deltim is set to a value of:

- 1 s for *P*-wave phases,
- 3 s for *S*-wave phases,
- 5 s for *Lg* phases.

All phases in ARRIVAL_EX that still have *deltim* ≤ 0.0 are now removed and written to REMOVED_ARRIVAL. Further, any *P*-wave phases at distances greater than 114º and any *S*-wave phases at distances greater than 19º are removed. Finally, all *delta* parameters are recomputed and updated in the ASSOC table.

Next all arrivals at stations in the MISSING_SITE table are removed from ARRIVAL_EX and written to REMOVED_ARRIVAL. This step may not be necessary in future runs of the merge when all site information becomes reconciled.

The quality of each arrival is now assessed by evaluating the travel-time residual normalized to expected fluctuations in travel-time prediction and measurement error. This is done using the PHASE VALIDATOR code which computes a theoretical arrival time and model uncertainty for each *arrival.* The code uses a *tau-p* formulation and the theoretical travel time and travel time uncertainty from the *iasp91* model are used with the observed travel time, pick error (*deltim*), and an assumed origin time uncertainty of 2 s to compute a normalized error. If the normalized error is greater than 3 (*i.e.*, the residual is a 3σ outlier) then the *arid* is removed. This process essentially removes gross outliers from the dataset.

Next, we rename phases in the ARRIVAL_EX and EVENT_ARRIVAL_ASSOC tables for consistency and to allow for more easy preparation of KBCIT input files to compute correction surfaces. The following phase names are checked and renamed by distance range if necessary:

- *P,*
- *Pg,*
- *Pn,*
- *Sg,*
- *Sn,* and
- *Lg*.

For example, *P*-waves follow these guidelines:

- at *deltas* of 0.0º ≤ 1.4º the phase is *Pg*,
- for *deltas* 1.4º to 15.0º the phase is *Pn*, and
- at *deltas* greater than 15.0º the phase is *P.*

The final QC step performed on the arrivals is the merging. This is the selection of the best arrival for every *evid-sta-phase* tuple. A unique *arid* is chosen based on author ranking, and the *is_preferred* field in the ARRIVAL_EX table is set to '*y*'. *Arids* not chosen remain in the ARRIVAL_EX table but their *is_preferred* field is set to '*n*'. These picks are not removed to REMOVED__ARRIVAL because they have passed all other QC tests and are essentially good data. We now have the final set of arrivals with which to produce GT locations.

Because many arrivals have now been removed for various reasons, we check to see that every origin solution found in the GT_EPI_EX table that has a *method* string indicating a BMEB solution still meets its GT criteria. The defining ASSOC entries for each origin are evaluated against the BMEB criteria for that GT level (5, 20, or 25). The number of stations at local, regional, and teleseismic distances are checked as well as the primary and secondary azimuthal gaps. Events that do not pass the criteria are written to the RECALC_ORIGIN_RESOLVE table.

Note, that in many cases the *evid* will already be in the RECALC_ORIGIN_RESOLVE table because the event had solutions in common between the labs. In this case there is no action.

We relocate all BMEB events in RECALC_ORIGIN_RESOLVE using arrival gathers appropriate for the GT-level (*e.g.*, only stations within 250 km are used for GT5). If the new location meets its BMEB criteria, then we write the new ORIGIN_EX, ORIGERR, ASSOC, PREDICT_TT, and GT_EPI_EX rows and move the *evid* and *epicenterid* to the RESOLVED table. At this point we also review by hand any events in the MANUAL_RESOLVE table, reconcile them, and write the corresponding rows to the RESOLVED table.

Next we baseline the origin times of events in the RESOLVED table having depths <= 35 km. We calculate a constrained origin solution where *lat, lon, depth* remain unchanged and a new origin time is computed using only teleseismic *P* phases (must have minimum of 10 *P* phases). For each baselined origin, a row is written in the CAL_ORIGIN table. Rows are also written to ORIGIN_EX, ORIGERR, ASSOC, and PREDICT_TT. Baselining addresses two issues: it makes the regional travel times consistent with teleseismic times, and since the *Pn* tomography code does not include relocating the events when inverting for the *Pn* velocities this step insures consistent origin times with which to measure *Pn* residuals. Note, that in this step about 2300 events are encountered that have nominal depths greater than 35 km but which have epicenters well outside of known regions of deep seismicity. These events are baselined using a fixed origin depth of 10 km.

Appendix 1: Merge Procedure Steps (Code-level)

The merge is made in the GTMERGE schema, and it is written in PL/SQL and Java. Each of the steps outlined above is presented in detail in this section along with the tables used and populated at each step. Note, the merge was done at LLNL and drives off many tables in the LLNL schema while using a groomed set of tables from LANL which contain all relevant information that was pre-prepared. The GTMERGE schema and process are designed with the ability to track any individual event or pick from beginning to end

COPY_INPUT_DATA

First, all relevant tables in the GTMERGE schema are initialized by the CLEAR_TABLES procedure to begin with a clean slate. Data is then copied from the LLNL and LANL source tables into the GTMERGE target tables assigning new *ids* in the process.

Specifically, each *evid* in LANLGT.PNGRAD_UNIGT_GT_EPI_FIN is written into GT_EPI_EX. For each *orid*, corresponding rows from LANLGT.PNGRAD_UNIGT_ORIGIN, LANLGT.PNGRAD_UNIGT_ORIGERR, LANLGT.PNGRAD_UNIGT_ARRIVAL, LANLGT.PNGRAD_UNIGT_ASSOC are retrieved and inserted into ORIGIN_EX, ORIGINERR, ARRIVAL_EX, and ASSOC respectively.

Next each *evid* in LLNL.GT_EPICENTER is written into GT_EPI_EX after first determining the best origin solution. As LLNL.GT_EPICENTER has multiple rows per event, *i.e.*, more than one *orid* for each *evid*, the best origin solution information as ranked by *gtlevel_km* and *method* (from GT_METHOD_RANK) must be determined before populating the event row in GT_EPI_EX. The best origin solution is found in LLNL.PREFERRED_ORIGIN and the associated fields are then copied from LLNL.GT_OT, LLNL.GT_DEPTH, and LLNL.GT_ETYPES into GT_TIME_EX, GT_DEPTH_EX, and GT_ETYPE_EX as well as GT_EPI_EX. For the best *orids* the remaining rows from LLNL.ORIGIN, LLNL.ORIGERR, LLNL.ARRIVAL, and LLNL.ASSOC are copied into ORIGIN_EX, ORIGERR, ARRIVAL_EX, and ASSOC respectively.

At the end of this process, all the *ids* (*orid, arid*, *epicenterid*, *depthid, etypeid,* etc.) except for *evid* will be new and unique to the target GTMERGE schema; *evids* are all set equal to -1 at this point as they cannot be assigned until after matching origin solutions have been determined in the next step. Additionally, remap columns are written that allow tracing of *IDs* from the GTMERGE schema back to the origin LANL and LLNL schema tables.

After all input data have been copied, the REMOVE_INCONSISTENT_EVENTS procedure removes any events for which there is an inconsistency of any: *lat, lon, depth,* or *time* between the GT_EPI_EX and ORIGIN_EX tables. Corresponding rows are deleted from: ORIGIN_EX, GT_EPI_EX, EVENT_ARRIVAL_ASSOC, GT_TIME_EX, GT_ETYPE_EX, GT_DEPTH_EX, and these *evids* are written to REMOVED_EVENT. With the current dataset, no such rows have been found, so REMOVED_EVENT is empty.

The following tables are used in procedure COPY_INPUT_DATA:

Tables needed:	Tables populated:
LANLGT.PNGRAD_UNIGT_.ARRIVAL	ARRIVAL_EX
LANLGT.PNGRAD_UNIGT_.ASSOC	ASSOC
LANLGT.PNGRAD_UNIGT_.GT_EPI_FIN	GT_DEPTH_EX
LANLGT.PNGRAD_UNIGT_.GT_ORIGERR	GT_EPI_EX
LANLGT.PNGRAD_UNIGT_.ORIGIN	GT_ETYPE_EX
LLNL.ARRIVAL	GT_TIME_EX
LLNL.ASSOC	ORIGERR
LLNL.ORIGIN	ORIGIN_EX
LLNL.ORIGINERR	REMOVED_EVENT
LLNL.GT_EPICENTER	
LLNL.EVENT_TYPES	
LLNL.GT_DEPTH	
LLNL.GT_OT	
LLNL.PREFERRED_ORIGIN	
GT_METHOD_RANK	

Note, tables with the '_EX' ending are 'extended' tables. This is simply the standard CSS format table with extra columns at the end to allow quick remapping back to original LANL or LLNL table information. This bypasses building an additional remap table that would slow down processing efficiency.

GROUP_INPUT_DATA

This procedure evaluates each GT solution in GT_EPI_EX and ORIGIN_EX and assigns it to one of the following types:

RESOLVED
MANUAL_RESOLVE
RECALC_ORIGIN_RESOLVE

Each row in GT_EPI_EX from LLNL is checked against the LANL GT to determine if it is the same event (*i.e.*, it has the same origin *time, lat, lon* within given tolerances). Those events that have no match are unique to LLNL and are written to the RESOLVED table. At this time a new *evid* is assigned, and is added to EVENTS_TO_CALIBRATE.

If the events match, then after assigning the *evids,* the two GT_EPI_EX rows are ranked by *gtlevel_km* and *method* to determine the best solution. If one has lower (better) *gtlevel_km* then it is selected and written to RESOLVED while the other GT_EPI_EX row is written to REMOVED_ GT_EPI. If both rows have the same *gtlevel_km*, then the GT *method* is checked next. If both *methods* are BMEB (Bondar-Myers-Engdahl-Bergman) then the *evid* is written to the RECALC_ORIGIN_RESOLVE table and will be resolved by relocation using all arrivals from both labs.

Those GT_EPI_EX rows in common for which the *gtlevel_km* is the same and for which neither *method* is BMEB and with epicenter separation <= the *gtlevel_km* are resolved by moving the second GT_EPI_EX row to the REMOVED_GT_EPI table and adding the EPICENTERID and EVID of the first to the RESOLVED table. If the epicenter separation is greater than the *gtlevel_km*, the *evid* is written to the MANUAL_RESOLVE table.

This procedure is then repeated by checking each unresolved LANL GT_EPI_EX row against each unresolved LLNL GT_EPI_EX row, thereby identifying all unique LANL events.

We now load all other possible arrivals not already found in ARRIVAL_EX and ASSOC using procedure ARRIVAL_LOADER.LOAD.ALL_OTHER_ARRIVALS. The procedure loads all other *arrivals* that are linked to an *event* in the GTMERGE schema but are not linked to any specific *origin* in the GTMERGE schema. We now have a complete compilation of arrivals, even those with no corresponding ASSOC rows.

A series of cleanups are now applied to all the arrivals using procedures MAKE_STANDARD_PHASES NAMES, REMOVE_DISCREPANT_ARRIVALS, and REMOVE_UNWANTED_PHASES:

- Make phase names standard between both labs (*e.g.*, *P** is renamed to *Pb)*
- Remove *arrivals* with times before the associated origin time minus the factor origin_time_padding.
- Remove *arrivals* with times greater than max_allowable_travel_time after the origin time.
- Remove phases not found in the ALLOWABLE_PHASE table. This contains 21 phase names: *P, Sg, PcP, sP, Pdif, SmS, PmP, pP, pPn, sPg, S, sSn, sPn, pS, Pg, Lg, Sb, Pb, sS, Sn, Pn.*

Arrivals removed at this stage are assumed to have been mis-associated or at least are at stations too distant to be useful. These arrivals are deleted from ARRIVAL_EX and written to REMOVED_ARRIVAL.

The SITE table is now populated by checking the *sta* for every row in ARRIVAL_EX (For site information, we are using an experimental SITE table populated with data that are consistent among NEIC, ISC, LANL, IRIS, and LLNL). Any required station not found in this table is written to MISSING_SITE along with the number of affected phases.

The following tables are used in procedure GROUP_INPUT_DATA:

Tables needed:	Tables updated or populated:
ALLOWABLE_PHASE	ARRIVAL_EX
ARRIVAL_EX	ASSOC
GT_EPI_EX	EVENTS_TO_CALIBRATE
GT_METHOD_RANK	EVENT_ARRIVAL_ASSOC
GT_TIME_EX	GT_DEPTH_EX
LLNL.ARRIVAL	GT_EPI_EX
LLNL.ASSOC	GT_ETYPE_EX
LLNL.GT_EPICENTER	GT_TIME_EX
LLNL.EVENT_ARRIVAL_ASSOC	MANUAL_RESOLVE
LLNL.PHASE_DESC	MISSING_SITE
LLNL.PHASE_MAP	ORIGIN_EX
ORIGIN_EX	RECALC_ORIGIN_RESOLVE
STATION_METADATA.ALTERNATE_STATION	REMOVED_ARRIVAL
STATION_METADATA.MERGE_SITE_CSS	RESOLVED
STATION_METADATA.POTENTIAL_ALT_STATION	SITE

APPLY_ARRIVAL_CLEANUPS

A series of cleanups are now applied to all arrivals in the ARRIVAL_EX table. First, the procedure SETDEFAULTDELTIMVALUES assigns appropriate default *deltim* values if they are not already set. For each *arid*, if it's *phase* is found in the ALLOWABLE_PHASE table, and it's *deltim* ≤ 0.0 then the *deltim* is set according to default_p_deltim, default_s_deltim, and default_lg_deltim parameters. REMOVE_UNUSABLE_PHASES then removes all phases in ARRIVAL_EX that still have *deltim* ≤ 0.0 and writes them to REMOVED_ARRIVAL.

Next, the procedures REMOVE_DIFFRACTED_P and REMOVE_TELESEISMIC_S remove any phase in the list: *P, Pb, PcP, PmP, pP, sP; S, Lg ,Sb ,Sn, Sg, pS, sS* from ARRIVAL_EX which is beyond the P_cutoff_delta and S_cutoff_delta distances respectively. Removed *arids* are written to REMOVED_ARRIVAL. Arrivals in the EVENT_ARRIVAL_ASSOC tables are similarly checked (their *delta* is computed on the fly from associated ARRIVAL_EX, GT_EPI_EX, and SITE table rows) and removed as needed. Now every *arid* in ARRIVAL_EX has been evaluated for it's phase criteria.

Finally, all *deltas* are recomputed and updated in the ASSOC table in procedure UPDATE_ASSOC_DELTA.

The following tables are used in procedure APPLY_ARRIVAL_CLEANUPS:

Tables needed:	Tables updated or populated:
ALLOWABLE_PHASE	ARRIVAL_EX
ARRIVAL_EX	ASSOC
ASSOC	REMOVED_ARRIVAL
EVENT_ARRIVAL_ASSOC	
GT_EPI_EX	
ORIGIN_EX	
SITE	

REMOVE_NO_SITE_ARRIVALS

This procedure simply removes rows from ARRIVAL_EX which have a station found in the MISSING_SITE table and writes them to REMOVED_ARRIVAL.

The following tables are used in procedure REMOVE_NO_SITE_ARRIVALS:

Tables needed:	Tables updated or populated:
ARRIVAL_EX	ARRIVAL_EX
MISSING_SITE	REMOVED_ARRIVAL

PHASE_VALIDATOR

This Java code removes discrepant arrivals by comparing travel-time error statistics to a removal threshold. The statistic is the travel-time residual normalized by expected fluctuations in travel-time prediction and measurement error. For each *arrival* in ARRIVAL_EX, the residual is computed as the observed travel time minus the theoretical iasp91 travel. This is normalized by an error term that includes the model error, origin time uncertainty, and pick uncertainty. The origin solution from ORIGIN_EX or else information from GT_EPI_EX and GT_TIME_EX (for unassociated arrivals) and SITE information are used in determining the observed travel-time.

For each arrival, the theoretical travel time (T_{iasp}) and travel time uncertainty (E_{iasp}) from the *iasp91* model are computed using a *tau-p* formulation. We include as an additional source of error the uncertainty in the origin time. This is the E_{orig} term which is set to a constant value of 2.0 seconds. We then take the observed travel time (T_{obs}) and pick error (*deltim*) and compute the following:

$$\frac{(T_{obs} - T_{iasp})}{\sqrt{[(E_{iasp})^2 + (deltim)^2 + (E_{orig})^2]}}$$

If this is <=3 then we keep the arrival, if not we remove it and write it to the REMOVED_ARRIVAL table.

Currently in this PHASE_VALIDATOR procedure we are not renaming any phases (*e.g.*, *Pg* does not become *Pn*).

The following tables are used in procedure PHASE_VALIDATOR:

Tables needed:	Tables updated or populated:
ARRIVAL_EX	ARRIVAL_EX
GT_EPI_EX	REMOVED_ARRIVAL
GT_TIME_EX	
ORIGIN_EX	
SITE	

RENAME_PHASES

This procedure is part of the ARRIVAL_CLEANER package and is run in the PL/SQL environment. The following phase names are checked, and changed if necessary, for consistency and to allow for more easily preparing KBCIT input files to compute correction surfaces for: *P, Pg, Pn, Sg, Sn, Lg.*

For all *evids* in GT_EPI_EX, the *phase* and *delta* for each *arid* in ARRIVAL_EX and EVENT_ARRIVAL_ASSOC is checked and renamed following these rules:

- 0.0º ≤ 1.4º phase is *Pg*
- 1.4º to 15.0º phase is *Pn*
- 15.0º < greater phase is *P*
- 0.0º ≤ 1.46º phase is *Sg*
- 1.46º to 19.0º phase is *Sn*

If there is an *Sg* phase with a delta greater than 1.46º then it becomes *Lg.*

The following tables are used in procedure RENAME_PHASES:

Tables needed:	Tables populated:
ARRIVAL_EX	ARRIVAL_EX
EVENT_ARRIVAL_ASSOC	ASSOC

GT_EPI_EX	
SITE	

MERGE_ARRIVALS

This procedure identifies the best arrival in ARRIVAL_EX for each *evid-sta-phase* tuple for which there is more than one arrival, and marks those arrivals as 'preferred'.

The basic strategy is to use the global author rank table ARRIVAL_AUTH_RANK_SINGLE to determine which arrival to retain. In cases where one or more of the competing arrivals is from LLNL, we also use the LLNL.PREFERRED_ARRIVAL and LLNL.POOR_QUALITY_ARRIVAL in making the determination. ARRIVAL_AUTH_RANK_SINGLE is used to determine which arrival to retain.

The following tables are used in procedure MERGE_ARRIVALS:

Tables needed:	Tables updated or populated:
ARRIVAL_EX	ARRIVAL_EX
ARRIVAL_AUTH_RANK_SINGLE	
EVENT_ARRIVAL_ASSOC	
EVENTS_TO_CALIBRATE	
GT_EPI_EX	
LLNL.ARRIVAL_AUTH_RANK_SINGLE	
LLNL.POOR-QUALITY_ARRIVAL	
LLNL.PREFERRED_ARRIVAL	

CHECK_BMEB

This procedure examines every origin solution found in the GT_EPI_EX table that has a *method* string indicating a BMEB solution. The defining ASSOC entries for each origin are evaluated against the BMEB criteria, for that GT level (5, 20, or 25). The number of stations at local, regional, and teleseismic distances are checked as well at the primary and secondary azimuthal gaps. Also, solutions that have more than just the first-arriving *P* phase set to defining are not allowed. Events that do not pass the criteria have their *evid* written into RECALC_ORIGIN_RESOLVE.

The following tables are used in procedure CHECK_BMEB:

Tables needed:	Tables updated or populated:
ASSOC	RECALC_ORIGIN_RESOLVE
GT_EPI_EX	

RECALC_BMEB

This procedure drives off the RECALC_ORIGIN_RESOLVE table. For each *evid* in that table, the procedure brings together first-arriving *P* appropriate for the putative GT level and calculates a new origin solution. This is then examined to determine whether it in fact meets the criteria. If so, a new GT_EPI_EX row is created (along with ORIGIN_EX, ASSOC, ORIGERR, and PREDICT_TT rows). The *evid* is then removed from RECALC_ORIGIN_RESOLVE and placed in the RESOLVED table. If the new solution does not meet the criteria then rows are written to FAILED_BMEB_RECALC and to FAILED_BMEB_RECALC_DETAIL and the row is removed from RECAL_ORIGIN_RESOLVE. In either case, the original GT_EPI_EX rows are removed and placed in REMOVED_GT_EPI.

The following tables are used in procedure RECALC_BMEB:

Tables needed:	Tables updated or populated:
ARRIVAL_EX	ARRIVAL_EX
ASSOC	ASSOC
EVENT_ARRIVAL_ASSOC	EVENTS_TO_CALIBRATE
GT_EPI_EX	EVENT_ARRIVAL_ASSOC
	FAILED_BMEB_RECALC
	FAILED_BMEB_RECALC_DETAIL
	GT_EPI_EX
	ORIGERR
	ORIGIN_EX
	PREDICT_TT
	RECALC_ORIGIN_RESOLVE
	REMOVED_GT_EPI
	RESOLVED

MANUAL RESOLVE

Four events had to be manually resolved because they had identical gt levels but differed in position by more than the limit of the gt level. These were resolved by choosing the solution from the lab responsible for the region in which the event occurred. Specifically:

EVID 16662 used LANL solution
EVID 53958 used LANL solution
EVID 6755 used LLNL solution
EVID 54320 used LLNL solution

TELESEISMIC ORIGINTIME BASELINER

This java program produces origin solutions constrained by epicenters from GT_EPI_EX and either *depth* from GT_DEPTH_EX or else by the *depth* associated with GT_EPI_EX through a linked origin in ORIGIN_EX.

All available teleseismic *P* phases were used in the solution. If the event did not have 10 or more teleseismic *P* picks then it was not used. Only shallow events were baselined. The depth cutoff used was 35 km. Several thousand events outside of regions with known deep seismicity had depths greater than 35 km. Since these depths are probably erroneous, we fixed the depth to 10 km for those events before baselining them.

These solutions were written into ORIGIN_EX, ORIGERR, PREDICT_TT, and ASSOC tables. Also, entries were made in the CAL_ORIGIN table.

The following tables are used in procedure ORIGINTIME_BASELINER:

Tables needed:	Tables populated:
GT_EPI_EX	ARRIVAL_EX
GT_DEPTH_EX	ASSOC
ORIGIN-EX	CAL_ORIGIN
	GT_EPI_EX
	ORIGERR
	ORIGIN_EX
	PREDICT_TT

REGIONAL ORIGINTIME BASELINER

This java program produces baselined origin solutions with origin times determined from traveltime residuals calculated using specific regional models. We used this code to process a subset of the GT that could not be baselined using the teleseismic baseliner because there were not enough teleseismic *P* observations.

Like the teleseismic baselining code, the regional baseliner fixes the origin solutions to have the same epicenter and depth as the GT solution being baselined. It then computes the origin time as a weighted average of the difference between the arrival times and predicted travel times using an appropriate regional model. The new origin time *O* is given as:

$$O = \sum_{i}^{N} \omega_i (a_i - T_i)$$

Where

$$\omega_i = \frac{1}{\sigma_i \sum_{j}^{N} \frac{1}{\sigma_j}} \quad \text{and} \quad \sigma_i = \sqrt{deltim_i^2 + ModelError_i^2}$$

We used the phases *Pn*, *Pg*, and *P* and required a minimum of three such observations to baseline an event. For European Arctic events we use the EA model. For the China region we used the LMV model, and for the YSKP region we used the RBHmod model. These models are available in the SNL_Tool_Root_200611d.

Appendix 2: GT Merge Schema

The GTMERGE schema and ER diagram can be found in the attached **gtmerge.DataModel** folder which contains several *.html* files which can be opened in a web browser. Open the **index.htm** file first and all subsequent links can be found in this master file.

Note, this may not work with the Safari browser, but has been tested successfully with the Internet Explorer, Firefox, and Mozilla browsers.

Appendix 3: Production of Merged SITE table for the GT Merge

Background

Information about seismic station position and installed instrumentation is to a greater or lesser extent, fundamental to all the processing done within the GNEM Program. However, despite the importance of accurate information about seismic stations, in practice it is difficult to obtain a compilation of station information that does not include errors. There are many sources for these errors including:

- Imprecise surveying/reporting by station operators
- Transcription errors
- Unrecorded station movements or equipment modifications.

The situation is complicated even more by the fact that many different compilations have been produced using different sources and different assumptions, and these compilations are inconsistent with one another.

In the past, we have dealt with inconsistencies on a case by case basis. When a problem was identified, we would "fix" the offending data in our SITE table and go on. While this approach was problematic in a number of ways, given the limitations of the CSS SITE table and our need to build out other parts of our infrastructure, it was judged to be the best we could do. As the labs and AFTAC have begun to coordinate more in the process of producing calibration products for monitoring purposes, the need for a unified, consistent SITE table has become more apparent. Producing and maintaining such a table by integrating and reconciling our individual SITE tables is an even more difficult undertaking than simply maintaining an internal-use-only SITE table. Mainly this is because of the need to resolve conflicts in a way that is traceable, reproducible, and with documented decisions/assumptions.

WG 7/8 have had an action item for the last two years to develop a mechanism for maintaining merged seismic metadata (SITE, SITECHAN, SENSOR, and INSTRUMENT) tables for use among the labs and at AFTAC. However, to our knowledge, little progress has been made in implementing a merge mechanism. The GT merge forced the issue with respect to the SITE table. The GT origins and associated arrival information produced independently by LANL and LLNL are referenced to two different SITE tables. While these tables are consistent over a large portion of their overlap, there are many inconsistencies that must be resolved in order to produce the GT merge data set. Therefore, as part of the GT merge effort, we developed a framework for merging SITE data from multiple sources in a way that is repeatable and traceable. This section describes our SITE merge process.

Site Merge Process

Our merge process is implemented in Java and in PL/SQL and uses a number of tables to track metadata about the merge process. The framework is still under development and does not yet support:

- Merging of array-specific data (membership, offsets to reference element, etc.)
- Temporary name changes by specific organizations

The merged SITE table contains data from the March 2007 NEIC and ISC station books, the current LLNL SITE table, a version of the LANL SITE table provided for use in the merge, and SITE information extracted from IRIS SEED volumes.

The codes allow for repeated contributions by the same author allowing, for example, updating of the merged SITE as new versions of the NEIC station book become available. Our approach to merging SITE data is to handle the position, elevation, operating epochs, and station movements separately. We take this approach because there is no guarantee that a particular contributor's information about a SITE will be uniformly better or worse than information from another source. Fields currently being merged are:

- ONDATE,
- OFFDATE,
- LAT,
- LON,
- ELEV,
- STANAME.

Fields are merged individually by means of an author rank table subject to over-rides.

When SITE data come into the system, they are placed into a multi-author site table (and supporting tables) that hold all the unmerged data. Before a new merge is executed, a process is run that identifies unresolved discrepancies (position discrepancies > 1km, elevation discrepancies > 100m). Any stations with unresolved discrepancies are added to appropriate discrepancy tables. Although the merge can continue without resolving the discrepancies, these stations will not become part of the merged SITE table.

Discrepancies can be resolved in one of two ways; either by making entries in a preferred position (or preferred elevation) table or by making entries in a rejected position (elevation) table. The reason column in each of these tables allows up to a 2000 character discussion of the reason for the decision. With this system, it is relatively easy to find out why a particular position or elevation was or was not used, and if better information becomes available it is easy to change the first decision and re-do the merge. The software also helps resolve position discrepancies by producing KML files that allow display in Google Earth of clusters of discrepant station position estimates.

Handling of alternate station codes is still somewhat rudimentary in this system. The NEIC station book lists over 500 such alternate codes. These are stored in a table and our NEIC and ISC parsers do not create entries in the multi-author SITE table for these codes. However, the contributing SITE tables have many alternate codes that are not specifically called out as such.

Currently, our system identifies candidate alternate codes by doing a pair-wise comparison of positions for stations in the multi-author SITE table. Candidates not already in the alternate code table are placed in a candidate table where they can be inspected manually.

We have used our SITE merging system to combine SITE information from the most recent NEIC and ISC station books, the current NNSA GNEM SITE tables, and the IRIS SITE table (derived from data-less SEED volumes minus temporary deployments and California stations). There are nearly 36000 entries in the multi-author site table which produce nearly 14000 merged SITE entries. There are 166 preferred positions, 41 preferred elevations, 55 rejected positions, and 514 rejected elevations. The position overrides were determined mostly through a combination of inspection in Google Earth and residual analysis using GT events. Most of the elevation overrides were arrived at by comparison of reported elevations with elevations computed using the gtopo30 elevation model.

Construction of STANAME in merged SITE.

The STANAME field in the merged SITE is constructed by concatenating the NAME and LOCATION fields in the MULTI_AUTH_SITE table. The specific row in MULTI_AUTH_SITE is chosen entirely by means of the SITE_AUTH_RANK table.

As an example, for the station RUE the query:

```
select sta, name, location_string, a.auth, rank
  from multi_auth_site a, site_auth_rank b
 where sta = 'RUE'
  and a.auth = b.auth
 order by rank
```

returns:

STA	NAME	LOCATION		AUTH	RANK
RUE	R|dersdorf	Brandenburg, Germany		neic	0
RUE	Ruedersdorf	Germany		isc	10
RUE	GRSN/GEOFON Station Ruedersdorf	Germany	GEOFON Program, GFZ Potsdam, Germany	iris	15
RUE	Temp GEOFON Station Ruedersdorf	Germany	GEOFON Program, GFZ Potsdam, Germany	iris	15
RUE		Ruedersdorf, Germany		llnl	20
RUE	R?dersdorf	Brandenburg, Germany		lanl	21

The merge software chooses the first entry by neic to create the STANAME for RUE resulting in ***R|dersdorf; Brandenburg, Germany.***

Construction of ONDATE and OFFDATE in Merged SITE

Construction of the ONDATE and OFFDATE fields is by selecting the non-null fields from the MULTI_AUTH_SITE table joined to the SITE_AUTH_RANK table. After selection of a candidate, the code checks for an entry in the LLNL_DATA_EPOCHS table for the station to see if there are any ARRIVAL or WAVEFORM data that precede an ONDATE or follow an OFFDATE just determined. If so, the DATE from LLNL_DATA_EPOCHS is used instead. This step could be enhanced by inclusion of similar epoch data from other organizations.

Construction of LAT and LON in Merged SITE

Construction of the LAT and LON fields in the merged SITE begins by looking for an entry in the PREFERRED_SITE_POSITION table for the station being merged. If one is found then that author's LAT, LON pair is used in the merged SITE table. For example the station GZT has an entry in the PREFERRED_SITE_POSITION table with the remark

> "Comparison of 17 P-residuals computed using GT origin solutions shows the ISC position has on average 2 seconds smaller residual than the neic or lanl positions."

Based on author rank alone, the NEIC position would have been used for this site. Instead, the ISC position is used.

If no PREFERRED_SITE_POSITION entry is found then the code selects non-null LAT,LON from the MULTI_AUTH_SITE table joined to the SITE_AUTH_RANK table ordered by RANK. The rows returned by this query are filtered to remove any by authors found in the REJECTED_SITE_POSITION table. For example, in processing the station WRAK, the NEIC position would be chosen in the absence of a rejected position. However, there is an entry in the REJECTED_SITE_POSITION table for ISM with the remark:

> "Based on inspection in Google Earth, the isc position is much closer to the city of Ismayilli which is the place name for the site."

Therefore, the ISC position ends up in the merged site table.

There are 143 entries in the PREFERRED_SITE_POSITION table with a remark of:

> "Author chosen based on rank to resolve discrepancy <= 15 km"

These stations really need more scrutiny. There was not enough time to do research on each of these stations because of constraints imposed by the GT merge deadline, but with possible discrepancies of up to 15 km, some of these have the potential to introduce significant biases.

Construction of Elevation in Merged SITE

Elevation is processed in a similar manner to position using PREFERRED_SITE_ELEVATION and REJECTED_SITE_ELEVATION tables. There is a little bit of additional processing, however. The NEIC elevation entries are often set to 0 even when other authors have non-zero values. Apparently 0 serves as some kind of default. So, when selecting elevation, the code first does a select for elevation values > 0 ordered by author rank. Only if that does not return any rows is the unrestricted selection attempted.

Construction of Move Dates in Merged SITE

The NEIC practice is to always report the current station position. For stations that have moved at some time, the move is recorded in comments. These comments have variable amounts of information ranging from "Station may have been moved?" to detailed information about the move. In the merge schema, the comments have been extracted into a STATION_MOVEMENT table. There are 56 entries by the NEIC in this table that have enough information to construct a new CSS row (move date, old lat, old lon optionally old elev). Move data from the LLNL and

LANL source tables has also been added to the STATION_MOVEMENT table. There are currently 102 LLNL rows and 42 LANL rows.

To construct move dates (new epochs) in the merged SITE table, the code first identifies moves in the STATION_MOVEMENT table for the same station by different authors. For each such move the code identifies any date discrepancy between pairs where the old positions are the same to four decimal places. These are put in the STAMOVE_DATE_DISCREPANCY table and are not used in producing additional CSS epochs. Currently there are two discrepancies (KIV, 2-days), (ANMO, 946-days). The remaining moves are considered valid and result in additional CSS epochs.

Handling of Alternate Stations and Potential Alternate Stations

As mentioned earlier, the merge process currently has no mechanism for handling obsolete codes and private (network/organizational) codes. Partly, this is because I have no information about the time periods for which these are in effect. In the case of potential alternate station codes there is even more ambiguity. For these codes, the gtmerge did a pairwise comparison of all the positions in the MULTI_AUTH_SITE table having different codes and stored the two codes and the separation in the POTENTIAL_ALT_STATION table (for separations < 1km). Where the separation is 0 they are clearly alternate codes. For larger separations they may or may not be alternate codes.

In the case of private network/organizational codes there is also a problem with code conflicts. That is, the private code may conflict with an NEIC code or may even be in conflict with other codes from the same organization. To use these codes it seems to me that a mapping table may be required that maps a private code for a certain organization over a certain time period to a code that does not conflict with other codes. That much would not be very difficult to do if I can get the period of applicability for each such code. However, to use a SITE table constructed in this fashion would require that all data being processed using the SITE table would first have to be mapped to the CODE used in the SITE table.

For purposes of the GT merge I took the easy way out and identified all arrivals with stations not found in the SITE table that had a match in either the ALTERNATE_STATION table or the POTENTIAL_ALT_STATION table. I then created new SITE entries using the ALTSTA for the code and all the rest of the site row from the matching station SITE row. This added 143 rows to the SITE table

Site Merge Schema

NEIC Tables

The site merge effort began as an effort to more fully-parse the NEIC station book as it is available on the web (http://neic.usgs.gov/neis/station_book/station_list.html). This form of the station book contains free-form comments with important details about the station operating history, and it was our intention to capture as many of these details as possible in a usable format in the database. The schema used to store information from the station book is shown below in Figure 1.

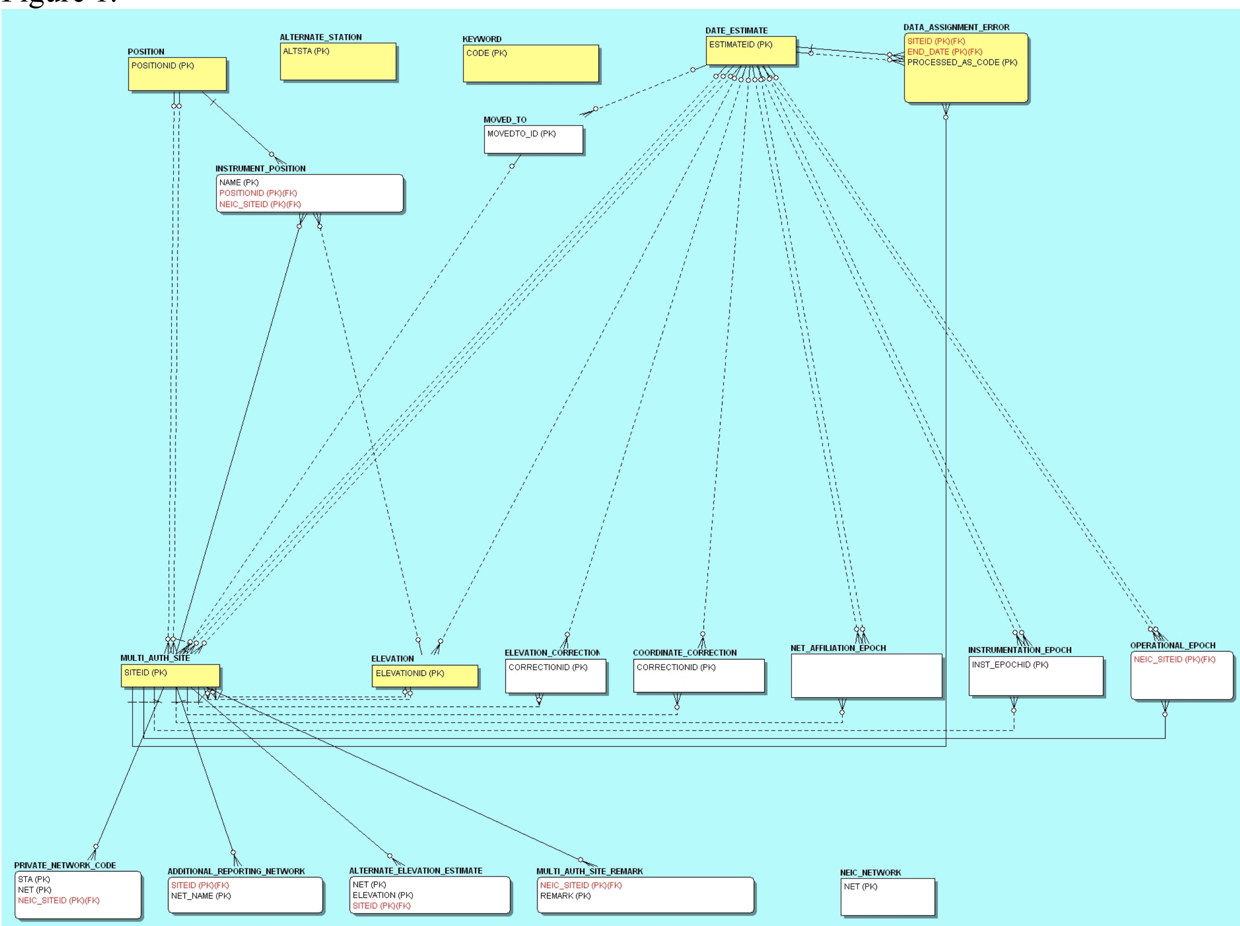


Figure 1. Schema tables used for parsing of NEIC station book.

Much of the information obtained from the station book is not directly relevant to the merge process. However, the tables shown in yellow in Figure XX are relevant to the merge and are discussed here.

The DATE_ESTIMATE table contains information about the dates of operating epochs, moves, and other important events in the station history. The reason that dates are placed in their own table is that the station book goes to some pains to describe how well known the date actually is. For example, date precision ranges from nearest decade to nearest day. Dates may also be described as questionable. In some cases the date is described as being either before or after the nominal date. Finally, dates may be described as planned or actual.

Similarly, positions have additional metadata that are stored in the POSITION table. There are remarks about the quality of positions, e.g. 'APPROXIMATE', 'GPS', etc. Occasionally, the datum used is mentioned, and coordinates may be characterized as 'PRELIMINARY', 'FINALIZED', etc.

The ELEVATION table contains elevations and additional metadata about elevations. Although breaking operational dates, positions, and elevations out of the SITE table makes the schema design more complex, it is a practical way to track all the metadata that may be associated with these quantities.

The ALTERNATE_STATION table contains station codes cross-referenced to the correct current station code for cases where two or more codes were mistakenly assigned to the same station at some time in the past. There are no SITE table entries for these alternate codes.

The KEYWORD table contains a list of NEIC reserved codes. It is intended that no station should use codes found in this table. Within the SITE merge process, data from non-NEIC sources are checked against this table to see if the code conflicts with a keyword.

The DATA_ASSIGNMENT_ERROR table is not used directly in producing the SITE merge. However, it should be useful in processing data that references the merges SITE table. This table tracks instances where an organization mistakenly processed a station using the wrong code. You would use it by joining your data on date, PROCESSED_AS_CODE and organization. Any rows in the result would need to have their station code corrected.

Merge Tables

Figure 2 shows the tables directly involved in producing the merged CSS SITE table. Many of these have been discussed previously, but in this section I present them in context.

The MULTI_AUTH_SITE table is the central table used in producing the merged SITE table. It contains information for all contributions for all stations available to the merge. A number of the columns are relevant only to NEIC contributions, but the columns:

- SITEID,
- STA,
- NAME,
- LOCATION_STRING,
- POSITION,
- ELEVATION,
- OPENDATE,
- CLOSEDATE,
- AUTH,
- STATUS

are used in the merge process. The SITEID key is a surrogate PK and (STA, AUTH) form an alternate key. The position, elevation, and date information are held in separate tables for reasons discussed in the previous section. When data from a specific source are re-loaded into this table, any instances of STA-AUTH for that author are replaced along with any child entries.

The SITE_AUTH_RANK table is the primary means of choosing a specific set of parameters from the MULTI_AUTH_SITE table for inclusion in the merged SITE table. The typical code path is to join the MULTI_AUTH_SITE table to the SITE_AUTH_RANK table on auth, order by RANK and then take the first row returned.

AUTH	RANK
neic	0
isc	10
iris	15
llnl	20
lanl	21
lanl_old	22
llnl_isc	30
pidc	40
unr	50
hel	60

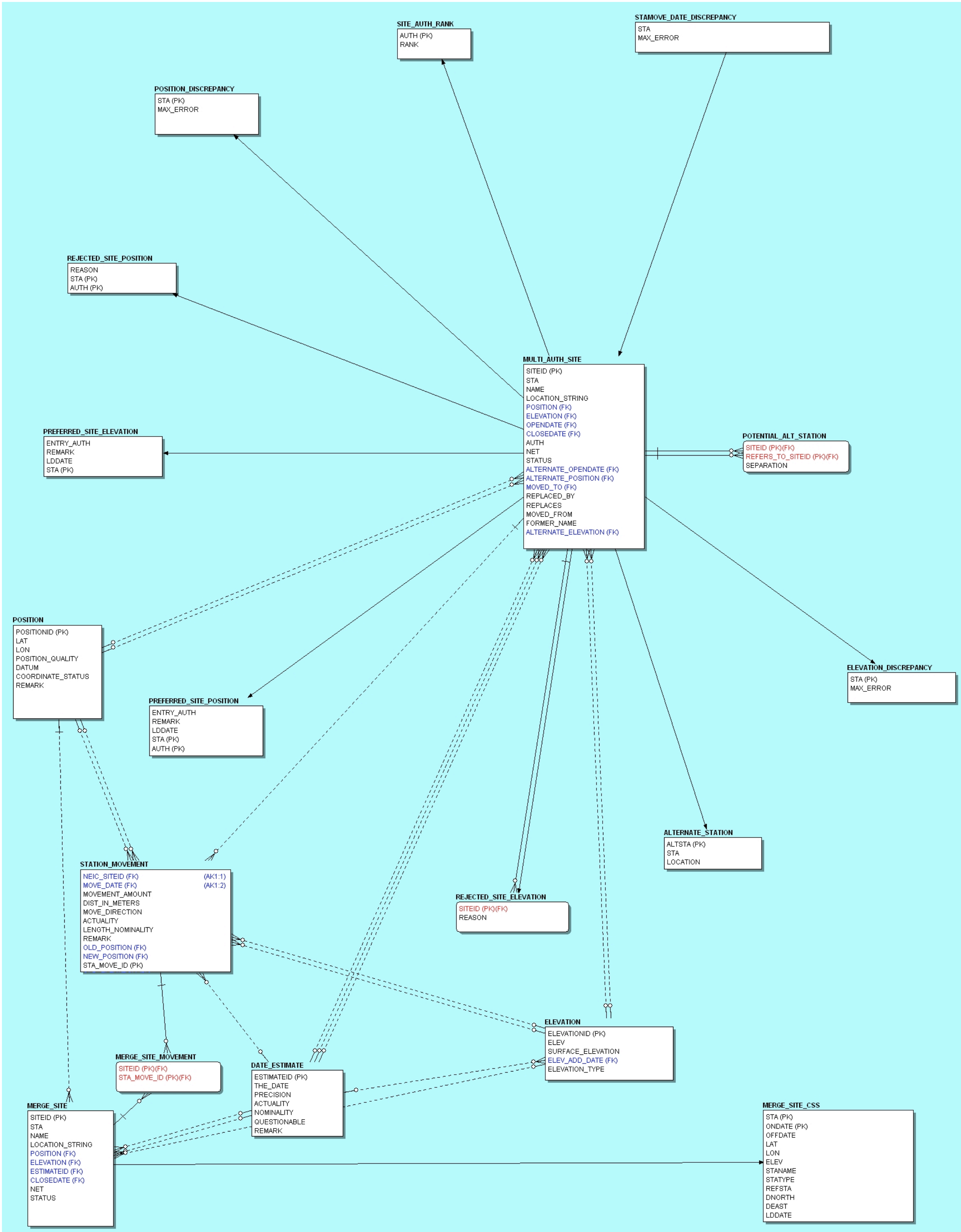


Figure 2. Tables used in producing the merged CSS SITE table.

The POSITION_DISCREPANCY table is populated early in the merge by software that looks at each set of rows in MULTI_AUTH_SITE with common authors and computes the distance between position estimates. Stations already in the PREFERRED_SITE_POSITION are not analyzed. Station-author combinations already in the REJECTED_SITE_POSITION table are also excluded. Subject to these restrictions, any station for which one or more of the separation estimates is greater than 1 km is added to the POSITION_DISCREPANCY. Any station found in the POSITION_DISCREPANCY table is rejected for inclusion into the MERGE_SITE table.

The ELEVATION_DISCREPANCY table is similar to the POSITION_DISCREPANCY table, but it stores information about stations for which there is a discrepancy in elevation of more than 100 meters. It is populated after the POSITION_DISCREPANCY table is populated. No stations in the POSITION_DISCREPANCY table are added to ELEVATION_DISCREPANCY. Also, no station entries found in the REJECTED_SITE_ELEVATION table are used in determining the maximum discrepancy.

The REJECTED_SITE_POSITION table contains information about a specific SITE position entry that has been found to be incorrect. During the merge, these data will be skipped.

The REJECTED_SITE_ELEVATION table contains information about a specific SITE elevation entry that has been found to be incorrect. During the merge, these data will be skipped.

The PREFERRED_SITE_ELEVATION table contains information about a specific SITE elevation entry that is considered to be the most reliable available. During the merge these values will be used regardless of the ranking of the entry author.

The PREFERRED_SITE_POSITION table contains information about a specific SITE position (lat, lon) entry that is considered to be the most reliable available. During the merge these values will be used regardless of the ranking of the entry author.

The STATION_MOVEMENT table contains information about position changes for stations. It is used in the merge to produce epochs for stations that have moved during their operating history.

The STAMOVE_DATE_DISCREPANCY table contains information about station movement pairs that share common before and after positions but that disagree on the date of the move. Only two such discrepancies have been recorded so far, and their presence in this table means that the move has not been propagated to the merged SITE table.

The MERGE_SITE_MOVEMENT table is an association table that links the MERGE_SITE table with the merged station movements that have been determined during the merge.

The MERGE_SITE table is an intermediate table that is the result of the merge in non-CSS form. Although not strictly necessary for the purpose of producing a merged CSS SITE table, it is related to ongoing experimentation with metadata handling.